通過テスト１０ 解答

10 抽象クラスとインタフェース

1. 問１

＜Factory クラス＞

```
package e1;
public abstract class Factory {
    abstract  public String toString();
}
```

＜MyFactory クラス＞

```
package e1;
public class MyFactory extends Factory{
    String product;
    int    price;
    public MyFactory(String product, int price){
        this.product =   product;
        this.price   =   price;
    }
    public String   toString(){
        return    product + ":" + price;
    }
}
```

＜Visible インタフェース＞

```
package e1;
public interface Visible {
    void   view();
}
```

＜FactoryViewer クラス＞

```
package e1;
public class FactoryViewer implements Visible{

    Factoryf;
    public FactoryViewer(Factory f){
        this.f    =   f;
    }
    public void  view(){
        System.out.println(f.toString());
    }
}
```

【解説】インタフェースの実装では，メソッドは public であることに注意する．

問 2

```
package e1;
public class Exec {
    public static void main(String[] args) {
        Factory  f  =  new MyFactory("BigCake", 100);
        Visible  v  =  new FactoryViewer(f);
        v.view();
    }

}
```

2. B, C

【解説】A はアクセス修飾子がデフォルトアクセスで protected より下げているので不可. B は正しい実装. C は引数型が double なのでオーバーロードである. C だけ書くと MyAdder クラスは抽象クラスを実装していないことになりコンパイルエラーになるが、B と共に C を書くのであればコンパイルエラーにならない。ここでは C も正解とする。D は抽象メソッドなので書けない．引数型は同じだが E は戻り値型が違うので不正なオーバーライド.

3. D

【解説】インタフェースで定義されるメソッドは public と明示されてなくても public なので，実装時には public アクセス修飾子を付けなければコンパイルエラーになる．なお，問題では Jumping インタフェースと Frog クラスに同じ型，同じ名前の変数 HEIGHT が宣言されている．Frog クラスで宣言された変数がインタフェース変数を隠ぺいする.インタフェース変数にアクセスしたい時は，Jumping.HEIGHT と書く.

4. B, F

【解説】A と C は CompareTo を実装していないので正しくない.B は正しい．D は run,compareTo を実装していない．E は CompareTo を実装していない．F は正しい.